Transactions Briefs

Architecture Design and Implementation of the Metric First List Sphere Detector Algorithm

Markus Myllylä, Joseph R. Cavallaro, and Markku Juntti

***Abstract*—Soft-output detection of a multiple-input–multiple-output (MIMO) signal pose a significant challenge in future wireless systems. In this paper, we introduce a soft-output modified metric first (MMF)-LSD algorithm for MIMO detection. We design a scalable architecture and address a method to decrease memory requirements. We provide implementation results for a spatial multiplexing (SM) system with four transmitted streams and with 16- and 64-quadrature amplitude modulation (QAM) on a 0.18-μm CMOS application specific integrated circuit (ASIC) technology. The MFF-LSD implementation is more efficient than the depth first (DF) -LSD in the crucial low signal-to-noise rate (SNR) region and the detection rate of the 64-QAM implementation is 39.2 Mbps@26 db with 48.2 kGEs complexity.**

***Index Terms*—Application-specific integrated circuit (ASIC), architecture, implementation, list sphere detector (LSD), multiple-input multiple-output (MIMO), soft-output detector.**

I. Introduction

Multiple-input–multiple-output (MIMO) techniques in combination with orthogonal frequency-division multiplexing (MIMO-OFDM) have been identified as a promising approach for high spectral efficiency wideband systems. The optimal maximum *a posteriori* (MAP) detector for MIMO system with forward error correction (FEC) coding is often too complex for systems with high order modulation. Suboptimal linear detectors [1] offer low complexity solutions, but have rather poor performance in correlated fading channels. A list sphere detector (LSD) [2] is a soft output variant of the sphere detector [3] that can be used to approximate the MAP detector with much lower computational complexity [2].

The SD algorithms are often divided according to their search strategy into the breadth first (BF), the depth first (DF), and the metric-first (MF) algorithms [4]. The BF algorithms [5] are implementation friendly, but have suboptimal performance. The DF algorithms [6] are more efficient in terms of visited nodes compared to the BF algorithms, but the algorithms have a variable search complexity and, thus, they are difficult to implement efficiently. The MF algorithms [4] are optimal in terms of the number of visited nodes, but require that the visited nodes are maintained in metric order to ensure the optimality, which requires the usage of memory and sorting [4]. Various sphere detector designs and implementations have been introduced, e.g., in [5], [7]–[9].

Manuscript received June 23, 2009; revised November 30, 2009. First published February 22, 2010; current version published April 27, 2011. This work was done in MITSE Project which was supported in part by Tekes, the Finnish Funding Agency for Technology and Innovation, Nokia, Nokia Siemens Networks, Elektrobit, and Uninord.

M. Myllylä and M. Juntti are with the Centre for Wireless Communications, University of Oulu, Oulu FIN-90014, Finland (e-mail: markus.myllyla@ee.oulu.fi; markku.juntti@ee.oulu.fi).

J. R. Cavallaro is with the Department of Electrical and Computer Engineering, Rice University, Houston, TX 77251-1892 USA (e-mail: cavallar@rice.edu).

Digital Object Identifier 10.1109/TVLSI.2010.2041800

In this paper, we introduce an implementation friendly soft-output modified MF (MMF) -LSD algorithm. We design an efficient and scalable architecture for the MMF-LSD and address the implementation trade-offs. We provide a scalable implementation for a spatial multiplexing (SM) system with up to 4 transmitted streams and 16- and 64-quadrature amplitude modulation (QAM) constellations on a 0.18-μm CMOS application specific integrated circuit (ASIC) technology. We present the synthesis and power results of the MMF-LSD implementation and compare it to a DF-LSD. As far as the authors know, there has been no other architecture designs or implementations of the MF-based detectors in the literature.

This paper is organized as follows. The signal model and the MMF-LSD algorithm are presented in Section II. The architecture is introduced in Section III, and the implementation tradeoffs are discussed in Section IV. The implementation results are introduced and discussed in Section V. The conclusions are drawn in Section VI.

II. MIMO Signal Detection

An OFDM-based SM system is considered with N_{T} transmit (TX) antennas and N_{R} receive (RX) antennas with the assumption $N_{\mathrm{R}} \geq N_{\mathrm{T}}$ and with QAM. A real signal model is assumed with the real dimensions $M_{\mathrm{T}} = 2N_{\mathrm{T}}$, $M_{\mathrm{R}} = 2N_{\mathrm{R}}$ and the real symbol alphabet $\Omega_{\mathrm{R}} \subset \mathbb{Z}$. The received signal can be expressed in the real domain as [3]

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \boldsymbol{\eta} \tag{1}$$

where the received signal vector $\mathbf{y}$, the transmit symbol vector $\mathbf{x}$, and the noise vector $\boldsymbol{\eta}$ are defined in the frequency domain. The noise elements of $\boldsymbol{\eta}$ are i.i.d. Gaussian with $2\sigma^2$ total power. The channel matrix $\mathbf{H} \in \mathbb{C}^{M_{\mathrm{R}} \times M_{\mathrm{T}}}$ contains complex Gaussian fading coefficients with unit variance.

An LSD [2] can be applied to solve a list of candidates $\mathcal{L} \in \mathbb{Z}^{N_{\mathrm{cand}} \times M_{\mathrm{T}}}$, where N_{cand} is the size of the candidate list, after the QR decomposition (QRD) of the channel matrix $\mathbf{H}$ as

$$\mathbf{x} = \arg \min_{\mathbf{x} \in \Omega_{\mathrm{R}}^{M_{\mathrm{T}}}} \|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{x}\|_2^2 \tag{2}$$

where $\mathbf{R} \in \mathbb{R}^{M_{\mathrm{R}} \times M_{\mathrm{T}}}$ is an upper triangular matrix with positive diagonal elements, $\tilde{\mathbf{y}} = \mathbf{Q}^{\mathrm{H}}\mathbf{y}$, and $\mathbf{Q} \in \mathbb{R}^{M_{\mathrm{R}} \times M_{\mathrm{R}}}$ is an orthogonal matrix. Due to the upper triangular form of $\mathbf{R}$ the values of $\mathbf{x}$ can be solved from (2) level-by-level using the back-substitution algorithm. Let $\mathbf{x}_i^{M_{\mathrm{T}}} = (x_i, x_{i+1}, \ldots, x_{M_{\mathrm{T}}})^{\mathrm{T}}$ denote the last $M_{\mathrm{T}} - i + 1$ components of the vector $\mathbf{x}$. The squared partial Euclidean distance (PED) of $\mathbf{x}_i^{M_{\mathrm{T}}}$ can be calculated as [7]

$$\begin{aligned} d(\mathbf{x}_i^{M_{\mathrm{T}}}) &= d(\mathbf{x}_{i+1}^{M_{\mathrm{T}}}) + |\tilde{y}_i - \sum_{j=i}^{M_{\mathrm{T}}} R_{i,j} x_j|^2 \\ &= d(\mathbf{x}_{i+1}^{M_{\mathrm{T}}}) + |b_{i+1}(\mathbf{x}_{i+1}^{M_{\mathrm{T}}}) - R_{i,i} x_i|^2 \end{aligned} \tag{3}$$

where $d(\mathbf{x}_{M_{\mathrm{T}}}^{M_{\mathrm{T}}}) = 0$, $b_{i+1}(\mathbf{x}_{i+1}^{M_{\mathrm{T}}}) = \tilde{y}_i - \sum_{j=i+1}^{M_{\mathrm{T}}} R_{i,j} x_j$, $R_{i,j}$ is the (i,j)th term of $\mathbf{R}$ and $i = M_{\mathrm{T}}, \ldots, 1$. The LSD output candidate list

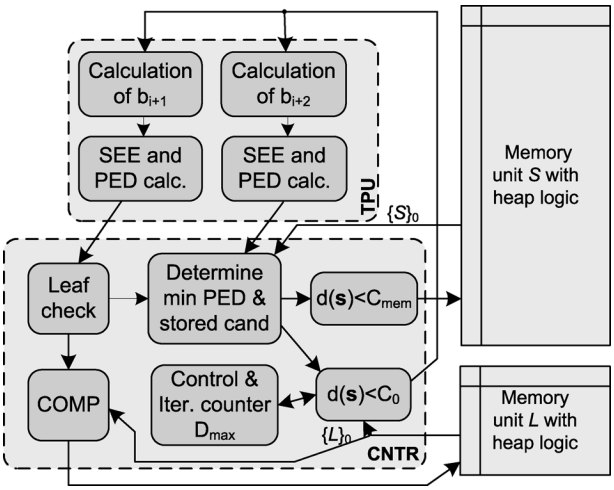

Fig. 1. Scalable architecture for the MMF-LSD algorithm.

is then used to approximate the soft outputs [2]. A computationally efficient max-log-MAP approximation of the log-likelihood ratio (LLR) of the kth transmitted bit b_k is calculated as [2]

$$L_D(b_k|\mathbf{y}) \approx \max_{\mathbf{x}\in\chi_{k,1}}\left(\frac{-d(\mathbf{x})}{2\sigma^2}\right) - \max_{\mathbf{x}\in\chi_{k,0}}\left(\frac{-d(\mathbf{x})}{2\sigma^2}\right) \tag{4}$$

where $\chi_{k,1} = \{\mathbf{x}|b_k = 1\}$ include the bit vectors in the candidate set $\mathcal{L}$ having $b_k = 1$.

We propose a MMF-LSD algorithm, which is a modification from the increasing radius (IR)-LSD in [10]. We include a maximum limit for the algorithm iterations $D_{\max}$ to fix the variable search complexity and transform the algorithm to be more suitable for implementation. The algorithm uses two memory sets: the final candidate memory $\mathcal{L}$ with the size of N_{cand} candidates and the partial candidate memory $\mathcal{S}$ with the size of $D_{\max}$ candidates. The MF search requires that the partial candidates are stored in $\mathcal{S}$ and the candidate with the minimum PED is extended on each iteration. We also propose to use a novel memory sphere radius C_{mem} to decrease the number of stored candidates and the complexity of the required min search. The extended partial candidates $\mathcal{N}$ are compared to the C_{mem} and stored to the memory $\mathcal{S}$ only if $d(\mathbf{s}) < C_{\text{mem}}$. We define C_{mem} based on the previously solved candidate(s) in the final list(s) with minimum ED $\min_{\mathbf{x}\in\mathcal{L}}(d(\mathbf{x}))$, which is then scaled with a determined radius scaling variable W_{R} to store only the potential partial candidates to the partial memory set $\mathcal{S}$. The minimum ED values can be averaged over time and frequency, i.e., OFDM subcarriers, and then the memory sphere radius can be written as

$$C_{\text{mem}} = W_{\text{R}} E[\min_{\mathbf{x}\in\mathcal{L}}(d(\mathbf{x}))]. \tag{5}$$

The impact of C_{mem} on complexity and performance will be illustrated with numerical examples in Section IV.

III. Architecture Design

A. MMF-LSD Algorithm

The MMF-LSD algorithm architecture, which operates in a sequential fashion, includes a tree pruning unit (TPU), a partial candidate memory unit, a final candidate memory unit, and a control logic (CNTR) unit as illustrated in Fig. 1.

1) TPU Unit: The TPU has two similar candidate extension modules, which execute the tree pruning for two nodes in parallel and can be divided into two sub-units as illustrated in Fig. 2. The first unit

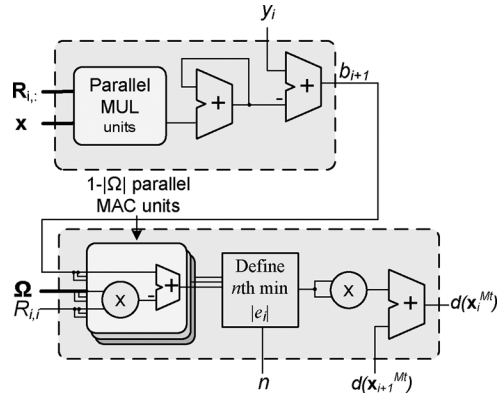

Fig. 2. Designed microarchitecture for the extension of the candidate.

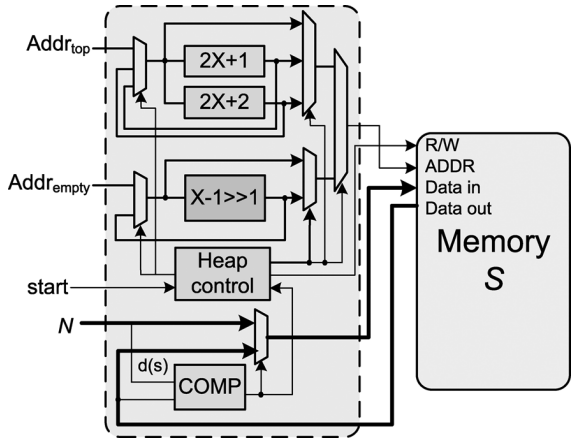

Fig. 3. Microarchitecture of the memory unit with heap logic.

calculates $b_{i+1}(\mathbf{x}_{i+1}^{M_{\text{T}}})$, which is the part of the PED calculation that is independent of the new symbol x_i, as in (3). The unit can be implemented with different levels of parallelism and/or pipelining for faster calculation of the multiplication (MUL) operations. The number of required multiplications is $M_{\text{T}} - i - 1$ and, thus, depends on the current layer i, where $i_{\max} = M_{\text{T}}$. Less than $M_{\text{T}}/2$ parallel MULs should be used in general to have an efficient implementation. The second unit executes the Schnorr–Euchner enumeration (SEE) and calculates the PED. The enumeration is designed in a modified fashion from the way presented in [3, (14)]. Instead of calculating the costly and high latency division operation, we calculate the absolute value in (3) with $|\Omega_{\text{R}}|$ different symbols x_i. The degree of parallelism should be decided depending on the slowest parallel unit in the whole MMF-LSD algorithm architecture to optimize the performance.

2) Memory Units: The memory units are designed as binary heap [11] data structures, which keep the stored elements in order according to the selected cost metric. The partial candidate memory set $\mathcal{S}$ is implemented as min-heap, where the elements $\mathcal{N}(\mathbf{s}, d(\mathbf{s}), n_2, i)$ are ordered so that the candidate with the minimum PED is always sorted to be at the top of the heap, and the final memory set $\mathcal{L}_{\text{F}}$ is implemented as max-heap. The storing of a new element requires a time complexity of $O(\log_2(k))$ in the worst case [11], where k is the size of the memory. The size of the partial candidate memory $\mathcal{S}$ is equal to $D_{\max}$ elements as at maximum the minimum candidate is removed and two candidates are added to the heap in each iteration. We modified the traditional heap sorting logic to limit unnecessary memory access. The possible new partial candidate(s) (child and father) are first compared to the minimum candidate, and if the candidate on the top of the heap has the minimum PED, the first stored candidate is located at the top of the heap and sorted via the down-heap operation [11]. Otherwise, the

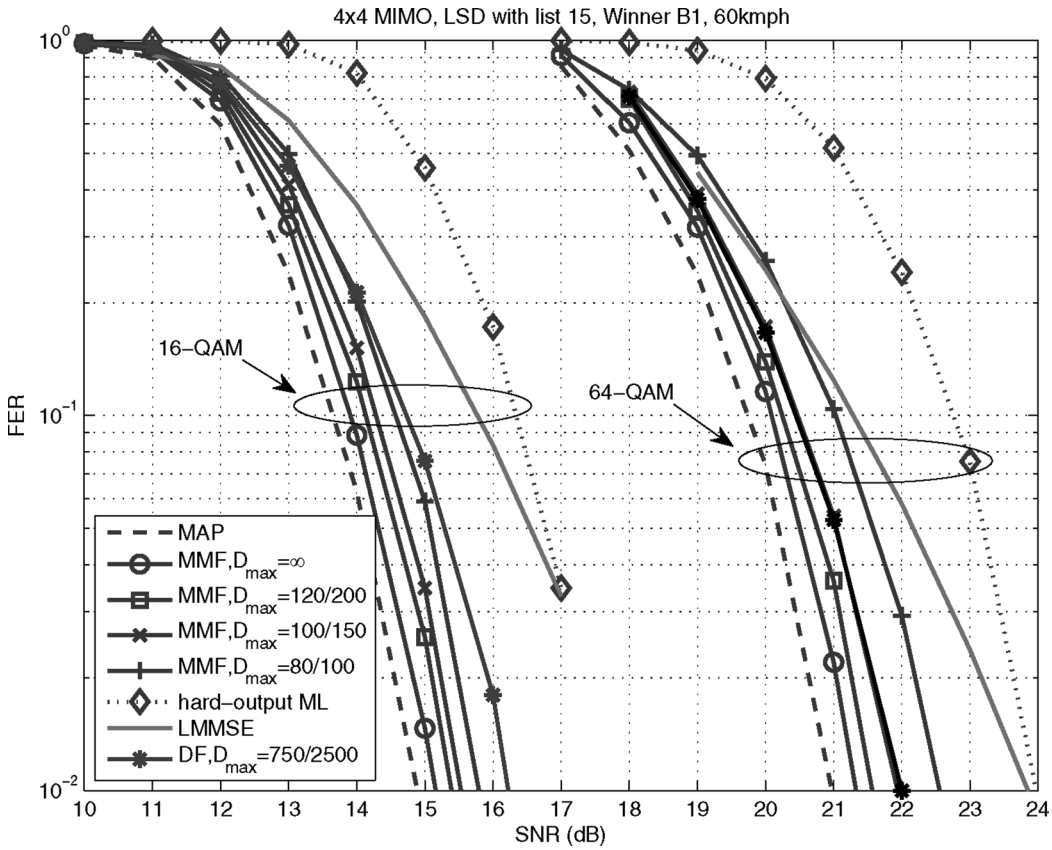

Fig. 4. FER versus SNR: Performance of the LSD based receivers in a 4×4 antenna system in Winner B1 channel.

new candidate is added to the next free memory address and the heap is sorted via the up-heap operation [11]. We also apply the memory sphere radius C_{mem} to decrease the amount of memory access as the updated candidates are discarded if $d(\mathbf{s}) < C_{\mathrm{mem}}$. The partial memory unit microarchitecture with up- and down-heap logic is illustrated in Fig. 3.

3) CNTR Unit and Data Flow: The control logic unit includes an iteration counter for the MMF-LSD algorithm and determines the candidates to be stored in the memory and to be used in the search in the next algorithm iteration. The candidate to be used in the TPU unit in the next iteration is determined as the candidate with minimum PED from the extended candidates $\mathcal{N}_c$ and $\mathcal{N}_f$, and the minimum candidate in partial memory $\mathcal{S}_0$. If either one of the extended candidates $\mathcal{N}_c$ or $\mathcal{N}_f$ is selected for the next algorithm iteration, $\mathcal{S}_0$ remains in the memory. Thus, unnecessary memory access is minimized as the candidates $\mathcal{N}_c$ and $\mathcal{N}_f$ are not directly stored in the memory. The extended partial candidate(s) to be stored in $\mathcal{S}$ are also conditioned with C_{mem} to minimize memory access. The data flow is designed to minimize the latency in one algorithm iteration by introducing parallel operations. The straightforward data flow would first extend the new candidates, then store them in memory units, and finally determine the new candidate for the next iteration. However, the data flow can be designed more efficiently to reduce the latency as follows: as the control logic unit determines the new candidate for the TPU at $D = 2$ and the stored candidates for the memory units from $D = 1$, the TPU and memory units are then executed in parallel, which decreases the latency significantly compared to the straightforward mapping.

B. LLR Calculation Unit

The soft output information $L_D(b_k)$ is calculated from the MMF-LSD algorithm output list $\mathcal{L}$ by using the max-log-MAP approximation as in (4). The microarchitecture can be divided into two main parts: the scaling of the ED values and the search for maximum values for each bit. The ED values in the candidate list $\mathcal{L}$ are scaled by multiplying them with the inverse of the noise variance $1/(2\sigma^2)$, i.e., a reciprocal division and a total of N_{cand} MUL operations are required. In practice 1–2 pipelined MULs should be used. The max-log-MAP approximation of $L_D(b_k)$ requires that all the N_{cand} ED values in the candidate list $\mathcal{L}$ are checked for each bit b_k in order to determine the maximum values for both bit counterparts. Thus, two sequential logic loops are required in the calculation with the final list index m and bit value index k. The latency of the loops can be decreased by applying parallel logic and/or pipelining to check multiple ED values or bits in parallel. It should be noted that the possibility for parallel implementation of the logic is a clear benefit of the max-log-MAP approximation compared to the log-MAP algorithm. The problem of inaccurate approximation can be compensated for by limiting the dynamic range of the output LLR variable [2].

TABLE I
AVERAGE NUMBER OF VISITED NODES AND EXECUTED HEAP OPERATIONS, AND PERFORMANCE LOSS IN MMF-LSD WITH C_{mem}

W_R	D_{avg}	**Up-heaps**	**Down-heaps**	**Loss** (dB)
$W_R^{16\mathrm{QAM}} = C_0$	77	1.09	1.45	0.0
$W_R^{16\mathrm{QAM}} = 2.0$	**47**	**0.25**	**0.72**	**0.1**
$W_R^{64\mathrm{QAM}} = C_0$	116	1.03	1.59	0.0
$W_R^{64\mathrm{QAM}} = 2.5$	**93**	**0.33**	**1.03**	**0.1**

C. Scalability

The MMF-LSD algorithm architecture can be used as such in SM systems with different antenna configurations and constellation sizes $|\Omega|$ as $N_{\mathrm{R}} \geq N_{\mathrm{T}}$. If some diversity method combined with SM scheme is applied with $N_{\mathrm{T}} > N_{\mathrm{R}}$, the receiver signal model should be modified accordingly, e.g., as in [12]. The limit for the number of algorithm

TABLE II
SYNTHESIS RESULTS OF THE ALGORITHMS: A SM SYSTEM WITH $N_{\mathrm{T}} = 4$, AND WITH 16- AND 64-QAM FOR 0.18-μm CMOS TECHNOLOGY

16/64-QAM	**Area** (mm^2)	**kGEs**	**Latency** (ns)	P (mW)
TPU	0.21/0.48	17.4/39.0	44/56	35.2/66.0
Part.mem.	0.06/0.06	4.6/5.2	40/44	10.8/12.0
Final mem.	0.02/0.02	1.4/1.7	20/20	3.7/4.4
CTRL	0.02/0.03	2.0/2.3	12/12	6.8/7.9
MMF alg.	0.31/0.59	25.4/48.2	56/68 per it.	56.5/90.3
DF alg.	0.13/0.27	10.6/22.0	52/64 per it.	24.9/38.4
LLR calc.	0.23	18.5	132/164	25.6

TABLE III
DETECTION RATES OF IMPLEMENTATIONS WITH DIFFERENT SNR

Resource	**MMF-LSD alg.**		**DF-LSD alg.**	
Mod. & SNR	16-QAM (@15/21dB)	64-QAM (@21/26dB)	16-QAM (@15/21dB)	64-QAM (@21/26dB)
ASIC	6.07Mbps/	3.80Mbps/	1.05Mbps/	1.01Mbps
$R_{\mathrm{det}}^{(\mathrm{asic})}$	31.7Mbps	39.2Mbps/	38.4Mbps	46.9Mbps

iterations $D_{\max}$ should be defined separately for different system configurations or according to the most complex supported configuration. A proper $D_{\max}$ value depends on the channel realization and on the search tree size, i.e., on the number of independent data streams and the constellation size $|\Omega|$. A larger tree size requires a higher $D_{\max}$ value. Memory resources of $D_{\max}$ elements are reserved for the memory unit $\mathcal{S}$ according to the highest supported system configuration. The amount of parallelism and pipelining in the TPU unit can be modified based on latency requirements. However, the TPU unit latency should be optimized to match the memory unit $\mathcal{S}$ and its logic, which are executed in parallel, for efficient implementation. The soft output LLR calculation unit can be used as such for different system configurations as it operates separately from the MMF-LSD algorithm. Multiple MMF-LSD algorithm units can be used in parallel to support higher data rate requirements. The scalability of the MMF-LSD is further illustrated with examples in Sections IV and V.

IV. IMPLEMENTATION TRADEOFFS

The MF algorithms as such are typically not suitable for low cost implementation. Therefore, we apply the limited search variable $D_{\max}$ and memory sphere radius C_{mem} in the MMF-LSD algorithm and LLR clipping in the LLR calculation to get a tradeoff between complexity and performance. It has also been shown that the detection order of the transmitted spatial streams affects the number of visited nodes [5]. Thus, we assume the use of sorted QRD (SQRD) [13] processing of the channel matrix $\mathbf{H}$ prior to the LSD algorithm, where the ordering of the spatial layers is included into a modified Gram–Schmidt decomposition process. The SQRD algorithm leads to close to optimal detection order so that the strongest signal is located at the top of the sphere search tree [13]. The SQRD of the channel matrix should be updated for all OFDM subcarriers within the channel coherence time. We do not focus on the SQRD implementation in more detail in this paper, but implementations of SQRD have been done, e.g., in [14].

We performed computer simulations to verify the feasible tradeoffs between performance and complexity of the MMF-LSD. A turbo coded MIMO-OFDM system was assumed with $N_{\mathrm{T}} = N_{\mathrm{R}} = 4$-, 16- and 64-QAM, and 512 subcarriers. A bit-interleaved coded modulation with $1/2$ rate turbo code with polynomial (13,15) was applied in a Winner B1 channel [15] with a user velocity of 60 kmph. The receiver includes a MMF-LSD with a list size $N_{\mathrm{cand}} = 15$ and a max-log-MAP turbo decoder with 8 iterations. A SQRD preprocessing is assumed in the LSD and the LLRs are limited to $|L_D(b_k)| < 8$.

The results are presented in frame error ratio (FER) versus signal-to-noise rate (SNR) γ in a Winner B1 channel [15] in Fig. 4. The performance of the MMF-LSD is also compared to that of the DF-LSD [2], linear minimum mean squared error (LMMSE) and ML detectors. We can see that the MMF-LSD algorithm works also with a limited maximum value $D_{\max}$ in the search and the performance loss of the MMF-LSD with $D_{\max}^{16\mathrm{QAM}} = 80$ and $D_{\max}^{64\mathrm{QAM}} = 150$ is approximately 0.6–0.8 dB compared to the unlimited search at 4% target FER, which we consider as an acceptable loss. The DF-LSD algorithm requires a much higher limit $D_{\max}^{16\mathrm{QAM}} = 750$ and $D_{\max}^{64\mathrm{QAM}} = 2500$ to obtain the same performance and the ML and LMMSE detectors lose 1–2 dB in performance. The required number of MFF-LSD algorithm iterations D decreases as SNR γ increases, because the algorithm is able to solve the detection problem with less visited nodes. The transceiver can be designed to operate at a certain target FER at desired SNR by selecting proper $D_{\max}$ values to satisfy the quality of service requirements. We also determined that a maximum of 12 and 15 bit fixed-point word lengths in the MMF-LSD result in very close to floating point performance with 16- and 64-QAM, respectively.

We also studied the affect of the memory sphere radius C_{mem} on the MMF-LSD performance and determined a proper value for W_{R} to be used. We studied the complexity reduction as the average number of executed iterations D_{avg} with $D_{\max}^{16\mathrm{QAM}} = 80$ and $D_{\max}^{64\mathrm{QAM}} = 150$, and the average number of required heap operations in the partial memory $\mathcal{S}$ in one subcarrier detection. The numerical results of the MMF-LSD with different C_{mem} values are listed in Table I. The performance of the MMF-LSD is not degraded significantly as the W_{R} value is selected to be large enough and only the potential partial candidates are stored to partial memory $\mathcal{S}$ with $W_{\mathrm{R}}^{16\mathrm{QAM}} \geq 2.0$ and $W_{\mathrm{R}}^{64\mathrm{QAM}} \geq 2.5$. As the C_{mem} is relative to the average minimum candidates over time and frequency, unnecessary resources, i.e., algorithm iterations, are not executed to obtain candidates with relatively high ED in the case of difficult channels. Thus, the average numbers of visited nodes decreases without loosing performance as shown in Table I. Also the average number of up- and down-heap operations are decreased, which significantly reduces the required memory access and the latency of the heap sorting.

V. IMPLEMENTATION RESULTS

The soft-output MMF-LSD algorithm was implemented for a SM system with $N_{\mathrm{T}} = 4$ with 16- and 64-QAM as the highest supported constellation, and with $D_{\max}^{16\mathrm{QAM}} = 80$ and $D_{\max}^{64\mathrm{QAM}} = 150$. The LLRs are calculated with max-log-MAP approximation. Both MMF-LSD algorithm implementations scale down for N_{T} and constellation sizes. The implementation of the MMF-LSD algorithm was targeted to a 0.18-μm CMOS ASIC technology using ANSI C++ language. The Mentor Graphics' Catapult C Synthesis tool was applied to produce the RTL description. The synthesis was done with Synopsys Design Compiler for 250 MHz frequency and the power usage was estimated with Synopsys Prime Power tool.

The detailed synthesis and power usage results are shown in Table II. The ASIC complexity is given in gate equivalents (GEs), where one GE corresponds to the area of a two-input drive-one NAND gate. The TPU unit is the most complex unit, while the others require only a minor part of the total resources. The TPU unit for 64-QAM is implemented with 2 and 4 parallel and pipelined MULs in the subunits, while the unit for 16-QAM is implemented with 2 and 2 MULs, to enhance the more demanding processing due to higher constellation and word lengths. The partial memory size is $D_{\max}$ candidate words and it is implemented with dual port memory to enhance the performance. The total power usage is $P = 56.5$ mW and $P = 90.3$ mW for the MMF-LSD algorithm with 16- and 64-QAM, respectively. The latency of a MMF-LSD algorithm iteration consists of the slowest parallel unit with the average

TABLE IV
SPHERE DETECTOR IMPLEMENTATION COMPARISON FOR 4 × 4 SYSTEM WITH CLOSE TO MAX-LOG PERFORMANCE

Algorithm and technology	STS, 0.25μm [7]	K-best K=5, 0.13μm [5]	K-best K=64, 0.13μm [8]	MMF-LSD, 0.18μm	
Constellation	16-QAM	16-QAM	64-QAM	16-QAM	64-QAM
Complexity (kGE)	56.8	97	280	25.4 + 18.5	48.2 + 18.5
Power (mW)	-	626	94	57 + 26	90 + 25
Max. Detection Rate (Mbps)	96 (Inf. bits)	106.6	8.57	31.7 and 121.2	39.2 and 146.3

number of required operations, the TPU unit, and the control logic. We also implemented a DF-LSD algorithm [2] to have a fair comparison to the MMF-LSD algorithm. The complexity of the MMF-LSD algorithm is more than double in GEs compared to the DF-LSD algorithm mainly due to the larger TPU unit, which enables two studied nodes in one iteration. The more sophisticated search of the MMF-LSD algorithm requires a bit more complex control logic and partial memory unit, but it also requires much less algorithm iterations on average. The LLR calculation unit is implemented with two MUL units and pipelined $N_{\mathrm{cand}} = 15$ parallel comparison units, and results are shown in Table II. The LLR calculation unit is used separately after the MMF-LSD algorithm search is executed, and, thus, the latency is different.

The detection rate R_{det} depends on the iterations D, which should be selected to meet the desired FER target with given channel and SNR. The detection rate of the MMF-LSD algorithm is listed and compared to the DF-LSD algorithm implementation with low SNR $\gamma = 15/21$ dB and with high SNR $\gamma = 21/26$ dB in Table III. The detection rate R_{det} of the MMF-LSD algorithm at low SNR is approximately 4–6 times that of the DF-LSD algorithm with approximately 2–3 times more complexity. The detection rate at high SNR is approximately the same because both detectors compute sufficiently good LLR approximations with only a minimum number of iterations. In practice, the low SNR scenario is more important, because the receiver has to be designed according to the worst case. The impact of $D_{\max}$ on R_{det} and FER can be derived from the results. The LLR calculation unit achieve a fixed rate of $R_{\mathrm{det}}^{(\mathrm{asic})} = 121.2/146.3$ Mbps for 16-/64-QAM with 18.5 kGE complexity, respectively. In an OFDM system, one unit can be used for multiple parallel LSD algorithm units. The delay of the detection is the combined latency of the MMF-LSD algorithm and LLR calculation unit for one subcarrier.

A. Comparison to Literature and Discussion

We compare the implementation detection rates at high SNR to other sphere detector implementations for 4×4 systems with 16- and 64-QAM constellations in Table IV. It can be seen that our implementation is competitive with the other implementations. Especially the complexity and power usage is lower compared to the K-best implementations. It can be also noted that the detection rate of our implementation is competitive to the other presented designs even though the main advantage of the MMF-LSD is at low SNR. The main limiting factor of our implementation for higher throughput is the latency of one algorithm iteration, and the TPU unit, which could be enhanced, e.g., by using more advanced ASIC technology.

VI. CONCLUSION

We considered detection based on the MF search strategy and introduced a soft-output MMF-LSD algorithm, which is more suitable for implementation. We introduced an architecture for the algorithm and showed that the main operations of the algorithm can be run in parallel and they are scalable to different system configurations with minor changes. We also introduced the memory sphere radius, which reduces the memory access requirements and decreases the average number of visited nodes. An optimized architecture was implemented as scalable for a SM system with up to 4 transmitted streams and 16- and 64-QAM constellations on a 0.18-μm CMOS ASIC. The results show that the MMF-LSD algorithm is more efficient compared to the DF-LSD at low SNR.

ACKNOWLEDGMENT

The authors would like to thank Mentor Graphics for the possibility to evaluate Catapult C Synthesis tool.

REFERENCES

[1] G. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs Tech. J.*, vol. 1, no. 2, pp. 41–59, Aug. 1996.

[2] B. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 389–399, Mar. 2003.

[3] M. O. Damen, H. E. Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Trans. Inform. Theory*, vol. 49, no. 10, pp. 2389–2402, Oct. 2003.

[4] J. Anderson and S. Mohan, "Source and channel coding: An algorithmic approach," *IEEE Trans. Commun.*, vol. 32, no. 2, pp. 169–176, Feb. 1984.

[5] Z. Guo and P. Nilsson, "Algorithm and implementation of the K-best sphere decoding for MIMO detection," *IEEE J. Select. Areas Commun.*, vol. 24, no. 3, pp. 491–503, Mar. 2006.

[6] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Math. Comput.*, vol. 44, no. 5, pp. 463–471, May 1985.

[7] C. Studer, A. Burg, and H. Bolcskei, "Soft-output sphere decoding: Algorithms and VLSI implementation," *IEEE J. Select. Areas Commun.*, vol. 26, no. 2, pp. 290–300, Feb. 2008.

[8] S. Chen, T. Zhang, and Y. Xin, "Relaxed K-best MIMO signal detector design and VLSI implementation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 3, pp. 328–337, Mar. 2007.

[9] B. Widdup, G. Woodward, and G. Knagge, "A highly-parallel VLSI architecture for a list sphere detector," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, France, Jun. 20–24, 2004, pp. 2720–2725.

[10] M. Myllyla, M. Juntti, and J. Cavallaro, "Architecture design and implementation of the increasing radius – List sphere detector algorithm," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2009, pp. 553–556.

[11] D. Knuth, *The Art of Computer Programming*, 3rd ed. Boston, MA: Addison-Wesley, 1997, vol. 3, Sorting and Searching.

[12] Y. Wu and C. Tellambura, "Low-complexity optimal detection for hybrid space-time block coding and spatial multiplexing," in *Proc. IEEE 64th Veh. Technol. Conf. (VTC)*, Montreal, QC, Canada, Sep. 2006, pp. 1–4.

[13] D. Wübben, R. Böhnke, V. Kühn, and K. Kammeyer, "MMSE extension of V-BLAST based on sorted QR decomposition," in *Proc. IEEE Veh. Technol. Conf. (VTC)*, Orlando, FL, Oct. 6–9, 2003, vol. 1, pp. 508–512.

[14] P. Luethi, C. Studer, S. Duetsch, E. Zgraggen, H. Kaeslin, N. Felber, and W. Fichtner, "Gram-Schmidt-based QR decomposition for MIMO detection: VLSI implementation and comparison," in *Proc. IEEE Asia Pacific Conf. Circuits Syst. (APCCAS)*, Nov.-Dec. 30–3, 2008, pp. 830–833.

[15] J. Salo, G. Del Galdo, J. Salmi, P. Kyösti, L. Hentilä, M. Milojevic, D. Laselva, P. Zetterberg, and C. Schneider, "MATLAB implementation of the WINNER Phase I channel model," 2005. [Online]. Available: https://www.ist-winner.org/phase_model.html